CLAIMS:

Sub A1

1. A configuration method of a multi-OS for running a plurality of operating systems on a single computer, comprising the steps of:

starting a first operating system (OS);

incorporating a processing procedure common to said plurality of operating systems as a device driver of said first operating system in order to arrange said processing procedure in an area shared by said plurality of operating systems;

switching control from an interrupt processing unit of said first operating system to an interrupt processing unit for said common processing procedure by means of said device driver; and

loading a second operating system into said computer by means of said device driver and starting said second operating system in a virtual address space different from said first operating system;

whereby said interrupt processing unit for said common processing procedure schedules execution of each operating system so that said plurality of operating systems run on said single computer without changing a processing procedure of said first operating sytem.

2. A configuration method of a multi-OS according to claim 1, wherein in order to prevent that an interrupt resource used by one operating system is accessed by other operating systems, said interrupt resource is reserved not to be accessed by said other operating

systems when said one operating system requires said interrupt resource.

3. A configuration method of a multi-OS according to claim 1, wherein when one operating system does not require an interrupt resource after said interrupt resource has been assigned to said one operating system, reservation of said interrupt resource for said one operating system is canceled so that said other operating systems can utilize said interrupt resource.

4. A configuration method of a multi-OS according to claim 1, wherein addresses of said common processing procedure and common data stored in a memory area shared by said plurality of operating systems are coincident even in a virtual address space of any operating system.

5. A configuration method of a multi-OS according to claim 1, comprising:

acquiring address information of a function group of an operating system to be monitored;

extracting an address of an error function called upon occurrence of a failure in the operating system from said address information;

writing an interrupt occurrence instruction in said address of said extracted error function;

causing said interrupt processing unit for said common processing procedure arranged in said area shared by said plurality of operating systems to notify occurrence of a failure to a monitoring operating system when the interrupt occurs; and causing said monitoring operating system to perform diagnosis or recovery of the failure in said operating system to be monitored in response to said notification.

6. A method of running a second OS on a computer in which a first OS is running, comprising the steps of:

incorporating a multi-OS management program for managing a plurality of OSs into a virtual address space of said first OS as a device driver of said first OS;

making accessible to said incorporated multi-OS management program from said second OS and making virtual addresses of said multi-OS management program coincident in virtual address spaces of both said first and second OSs;

loading a second OS program itself into said virtual address space of said second OS by execution of said multi-OS management program in said first OS;

switching execution of said multi-OS management program from said multi-OS management program in said first OS to that in said second OS; and

starting said second OS by execution of said multi-OS management program in said second OS.

7. A method according to claim 6, wherein said multi-OS management program incorporated into said virtual address space of said first OS is copied to a memory area shared by said first and second OSs, so that said virtual address of said multi-OS management program is coincident in both said virtual address spaces of said

first and second OSs.

8. A storage medium for storing therein a program for running a second OS on a computer in which a first OS is running, said program comprising the steps of:

incorporating a multi-OS management program for managing a plurality of OSs into a virtual address space of said first OS as a device driver of said first OS;

making accessible to said incorporated multi-OS management program from said second OS and making virtual addresses of said multi-OS management program coincident in virtual address spaces of both said first and second OSs;

loading a second OS program itself into said virtual address space of ~~said~~ second OS by execution of said multi-OS management program in said first OS;

switching execution of said multi-OS management program from said multi-OS management program in said first OS to that in said second OS; and

starting said second OS by execution of said multi-OS management program in said second OS.